Fiftieth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
October 1 - 5, 2012

Computation over Mismatched Channels

Nikhil Karamchandani*, Urs Niesen†, and Suhas Diggavi‡
*Dept. of EE, UCLA, Los Angeles, CA 90095, USA and Dept. of ECE, UCSD, La Jolla, CA 92093, USA
Email: nikhil@ee.ucla.edu
†Bell Labs, Alcatel-Lucent, Murray Hill, NJ 07974, USA
Email: urs.niesen@alcatel-lucent.com
‡Dept. of EE, UCLA, Los Angeles, CA 90095, USA
Email: suhasdiggavi@ucla.edu

***Abstract*—We consider the problem of distributed computation of a target function over a multiple-access channel. If the target and channel functions are *matched* (i.e., compute the same function), significant performance gains can be obtained by jointly designing the computation and communication tasks. However, in most situations there is *mismatch* between these two functions. In this work, we analyze the impact of this mismatch on the performance gains achievable with joint computation and communication designs over separation-based designs. We show that for most pairs of target and channel functions there is no such gain, and separation of computation and communication is optimal.**

I. Introduction

The problem of computing a function from distributed information arises in many different contexts ranging from auctions and financial trading to sensor networks. In order to compute the desired target function, communication between the distributed users is required. If this communication takes place over a shared medium, such as in a wireless setting, the channel introduces interactions between the transmitted signals. This suggests the possibility to harness these signal interactions to facilitate the task of computing the desired target function. A fundamental question is therefore whether by jointly designing encoders and decoders for computation and communication, we can improve the efficiency of distributed computation.

A. Summary of Results

In this paper, we explore this question by considering the computation of a function over a two-user multiple-access communication channel. More formally, the setting consists of two transmitters observing a (random) variable $\mathsf{u}_1 \in \mathcal{U}$ and $\mathsf{u}_2 \in \mathcal{U}$, respectively, and a receiver aiming to compute the function $a(\mathsf{u}_1, \mathsf{u}_2) \in \mathcal{W}$ of these variables. The two transmitters are connected to the destination through a multiple-access channel (MAC) with inputs $x_1, x_2 \in \mathcal{X}$ and output $\mathsf{y} = g(x_1, x_2) \in \mathcal{Y}$, where $g(\cdot, \cdot)$ describes the actions of the channel.

A straightforward achievable scheme for this problem is to separate the tasks of computation and communication: the transmitters communicate the values of u_1 and u_2 to the destination, which then uses these values to compute the desired target function $a(\mathsf{u}_1, \mathsf{u}_2)$. This requires the receiver to decode $2\log|\mathcal{U}|$ message bits. However, the MAC itself also computes a function $g(x_1, x_2)$ of two inputs x_1, x_2, creating the opportunity of taking advantage of the structure of $g(\cdot, \cdot)$ to calculate $a(\cdot, \cdot)$. This is trivially possible when $g(\cdot, \cdot)$ and $a(\cdot, \cdot)$ are *matched*, i.e., compute the same function on their inputs. In such cases, performing the tasks of computation and communication jointly results in significantly fewer bits to be communicated. Indeed, in the matched case only the $\log|\mathcal{W}|$ bits describing the function value are recovered at the receiver. This could be considerably less than the $2\log|\mathcal{U}|$ bits resulting from the separation approach. Naturally, in most cases the channel $g(\cdot, \cdot)$ and the target function $a(\cdot, \cdot)$ are *mismatched*. The question is thus whether we can still obtain performance gains over separation in this mismatched situation. In other words, we ask if in general the natural computation done by the channel can be harnessed to help with the computation of the desired target function.

We consider two cases: i) *One-shot communication*, where the MAC is used once, but the channel input alphabet $\mathcal{X}$ and output alphabet $\mathcal{Y}$ are allowed to vary as a function of the the domain $\mathcal{U}$ of the target function. In this case, performance is measured in terms of the scaling needed for the channel alphabets with respect to the message alphabet, i.e., how $|\mathcal{X}|, |\mathcal{Y}|$ grow with $|\mathcal{U}|$. This is closer to the formulation in the computer science literature. ii) *Multi-shot communication*, where the channel alphabets $|\mathcal{X}|, |\mathcal{Y}|$ are of fixed size, but the channel can be used several times. In this case, performance is measured in terms of the computation rate, i.e., how many channel uses are needed to compute the target function. This is closer to the formulation considered in information theory.

As the main result of this paper, we show that, even in the case when the channel $g(\cdot, \cdot)$ is *deterministic*[1], separation between computation and communication is essentially optimal for most pairs (a, g) of target and channel functions. In other words, the structural mismatch between the functions $a(\cdot, \cdot)$ and $g(\cdot, \cdot)$ is in general too strong for joint computation and communication designs to yield any performance gains.

We illustrate this with an example for one-shot commu-

[1]In joint computation and communication the aim is to take advantage of the structure of the function $g(\cdot, \cdot)$ computed by the channel to more efficiently compute the desired target function $a(\cdot, \cdot)$. The presence of channel noise can only make it more difficult to harness the channel for computation. Thus, by focusing on deterministic channels, we are considering the most favorable situation for joint computation and communication.

978-1-4673-4539-2/12/$31.00 ©2012 IEEE

nication. Assume that the variables $\mathsf{u}_1, \mathsf{u}_2$ at the transmitters take on a large range of values, say $|\mathcal{U}| = 2^{1000}$, and the receiver is only interested in knowing if $\mathsf{u}_1 \geq \mathsf{u}_2$, i.e., in a binary target function. Then for most MACs and one-shot communication, a consequence of Theorem 2 in Section III (illustrated in Example 3) is that the transmitters need to convey the *entire* values of $\mathsf{u}_1, \mathsf{u}_2$ to the destination, which then simply compares them. Thus, even though the destination is interested in only a *single* bit about $(\mathsf{u}_1, \mathsf{u}_2)$, it is still necessary to transmit $2\log|\mathcal{U}| = 2000$ bits over the channel.

More generally, Theorems 1 and 2 in Section III together demonstrate that for most functions separation of computation and communication is asymptotically optimal for most MACs. Example 4 illustrates that only for special functions like an equality check (i.e., checking whether $\mathsf{u}_1 = \mathsf{u}_2$) can we significantly improve upon the simple separation scheme. The technical ideas that enable these observations are based on a connection with results in extremal graph theory such as existence of complete subgraphs and matchings of a given size in a bipartite graph. These connections might be of independent interest.

Similarly, for multi-shot communication, where we repeatedly use a fixed channel, Theorem 4 in Section III shows that for most functions, the computation rate is necessarily as small as that for the identity target function describing the entire variables $\mathsf{u}_1, \mathsf{u}_2$ at the destination. In other words, for a given channel function separation of computation and communication is again optimal for most target functions. To prove this result, the usual approach using cut-set bound arguments is not tight enough. Indeed, Example 6 shows that the ratio between the upper bound on the computation rate obtained from the cut-set bound and the correct scaling derived in Theorem 4 can be unbounded. Rather, the structures of the target and channel functions have to be analyzed jointly.

These results show that, in general, there is little or no benefit in joint designs: computation-communication separation is optimal for most cases. We thus advocate in this paper that separation of computation and communication for multiple-access channels is not just an attractive option from an implementation point of view, but, except for special cases, actually entails little loss in efficiency.

B. Related Work

The problem of distributed function computation has a rich history, see, e.g., [1] and references therein. Early seminal work by Yao [2] considered interactive communication between two parties. Among several other important results, the paper showed that the number of exchanged bits required to compute most target functions is as large as for the identity function. Distributed function computation was also studied by Körner and Marton in [3], where it was shown that for the computation of the finite-field sum of correlated sources, linear codes can outperform random codes. Orlitsky and Roche [4] derived the rate required for reliable computation of the function $a(\mathsf{u}_1, \mathsf{u}_2)$ when u_2 is available at the decoder. Function computation over networks represented as graphs has recently been studied in [5]–[7] and references therein.

In most of these works, communication channels are represented as orthogonal point-to-point links. When the channel itself introduced signal interaction, as is the case for a MAC, there can be a benefit from jointly handling the computation and communication tasks as illustrated in [8]. Function computation over MACs has been studied in [9]–[12] and references therein.

There is some work touching on the aspect of structural mismatch between the target and the channel functions. In [13], an example was given in which the mismatch between a linear target function with integer coefficients and a linear channel function with real coefficients can significantly reduce efficiency. In [12], it was conjectured that, for computation of finite-field addition over a real-addition channel, there could be a gap between the cut-set bound and the computation rate. In [14], mismatched computation when the network performs linear finite-field operations was studied. To the best of our knowledge, a systematic study of channel and computation mismatch is initiated in this work.

C. Organization

The paper is organized as follows. In Section II, we formally introduce the questions studied in this paper. We present the main results along with illustrative examples in Section III. For brevity, we do not include any proofs here. All proofs can be found in [15].

II. Problem Setting and Notation

Throughout this paper, we use sans-serif font for random variables, e.g., u. We use bold font lower and upper case to denote vectors and matrices, e.g., $\boldsymbol{y}$ and $\boldsymbol{G}$. All sets are typeset in calligraphic font, e.g., $\mathcal{X}$. We denote by $\log(\cdot)$ and $\ln(\cdot)$ the logarithms to the base 2 and e, respectively.

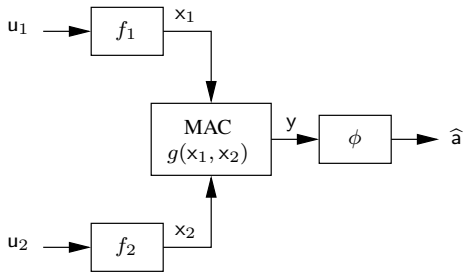

Fig. 1. Computation over a deterministic multiple-access channel. Each user i has access to an independent message u_i, and the receiver computes an estimate $\widehat{\mathsf{a}}$ of the target function $a(\mathsf{u}_1, \mathsf{u}_2)$ of those messages.

A discrete, memoryless, deterministic two-user multiple-access channel (MAC) consists of two *input alphabets* $\mathcal{X}_1$ and $\mathcal{X}_2$, an *output alphabet* $\mathcal{Y}$, and a deterministic *channel function* $g\colon \mathcal{X}_1 \times \mathcal{X}_2 \to \mathcal{Y}$. Given channel inputs x_1, x_2, the output of the MAC is

$$y \triangleq g(x_1, x_2).$$

Each transmitter $i \in \{1,2\}$ has access to an independent and uniformly distributed *message* $\mathsf{u}_i \in \mathcal{U}_i$. The objective of the receiver is to compute a *target function* $a\colon \mathcal{U}_1 \times \mathcal{U}_2 \to \mathcal{W}$ of the user messages, see Fig. 1.

Formally, each transmitter i consists of an *encoder* $f_i\colon \mathcal{U}_i \to \mathcal{X}_i$ mapping the message u_i into the channel input

$$\mathsf{x}_i \triangleq f_i(\mathsf{u}_i).$$

The receiver consists of a *decoder* $\phi\colon \mathcal{Y} \to \mathcal{W}$ mapping the channel output y into an *estimate*

$$\widehat{\mathsf{a}} \triangleq \phi(\mathsf{y})$$

of the target function $a(\mathsf{u}_1, \mathsf{u}_2)$. The *probability of error* is

$$\mathbb{P}\big(a(\mathsf{u}_1, \mathsf{u}_2) \neq \phi(\mathsf{y})\big).$$

Remark: We point out that this differs from the ordinary communication setting, in which the decoder aims to recover both messages $(\mathsf{u}_1, \mathsf{u}_2)$. Instead, in the setting here, the decoder is not interested in $(\mathsf{u}_1, \mathsf{u}_2)$, but only in the value $a(\mathsf{u}_1, \mathsf{u}_2)$ of the target function.

In the following, it will often be convenient to represent the target function $a(\cdot,\cdot)$ and the channel $g(\cdot,\cdot)$ by their corresponding matrices $\boldsymbol{A} = (a_{u_1,u_2}) \in \mathcal{W}^{\mathcal{U}_1 \times \mathcal{U}_2}$ and $\boldsymbol{G} = (g_{x_1,x_2}) \in \mathcal{Y}^{\mathcal{X}_1 \times \mathcal{X}_2}$, respectively. In other words,

$$a_{u_1,u_2} = a(u_1,u_2) \in \mathcal{W},$$
$$g_{x_1,x_2} = g(x_1,x_2) \in \mathcal{Y}.$$

For $n \in \mathbb{N}$, denote by $\boldsymbol{G}^{\otimes n}$ the n-fold use of the same channel matrix $\boldsymbol{G}$. In other words, the matrix $\boldsymbol{G}^{\otimes n}$ describes the actions of the (memoryless) channel $\boldsymbol{G}$ on the sequence

$$\big((x_1[1], x_2[1]), (x_1[2], x_2[2]), \ldots, (x_1[n], x_2[n])\big)$$

of length n of channel inputs.

Definition. A pair $(\boldsymbol{A}, \boldsymbol{G})$ of target and channel functions is *δ-feasible*, if there exist encoders f_1, f_2 and a decoder ϕ computing the target function $\boldsymbol{A}$ over $\boldsymbol{G}$ with probability of error at most δ.

Remark: We will often consider pairs $(\boldsymbol{A}, \boldsymbol{G}^{\otimes n})$, in which case the definition of δ-feasibility allows for coding over n uses of the channel $\boldsymbol{G}$.

Without loss of generality, we assume that the target function $\boldsymbol{A}$ has no two identical rows or two identical columns, since we could otherwise simply eliminate one of them. For ease of exposition, we will focus on the case

$$\mathcal{U}_1 = \mathcal{U}_2 = \mathcal{U},$$
$$\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{X}.$$

To simplify notation, we assume without loss of generality that

$$\mathcal{U} = \{0,1,\ldots,U-1\}, \qquad \mathcal{X} = \{0,1,\ldots,X-1\},$$
$$\mathcal{W} = \{0,1,\ldots,W-1\}, \qquad \mathcal{Y} = \{0,1,\ldots,Y-1\}.$$

Finally, to avoid trivial cases, we assume that all cardinalities are strictly bigger than one, and that $W \leq U^2$.

We denote by $\mathcal{A}(U,W)$ the collection of all target functions $a\colon \mathcal{U} \times \mathcal{U} \to \mathcal{W}$. Similarly, we denote by $\mathcal{G}(X,Y)$ the collection of all channels $g\colon \mathcal{X} \times \mathcal{X} \to \mathcal{Y}$. The next example introduces several target functions $\boldsymbol{A}$ and channels $\boldsymbol{G}$ that will be used to illustrate results in the remainder of the paper.

Example 1. We start by introducing three target functions $a(\cdot,\cdot)$.

- Let $\mathcal{W} = \mathcal{U} \times \mathcal{U}$. The *identity* target function is
$$a(u_1,u_2) \triangleq (u_1,u_2)$$
for all $u_1, u_2 \in \mathcal{U}$. Since we will refer to the identity target function repeatedly, we will denote it by the symbol $\boldsymbol{A}_I$.
- Let $\mathcal{W} = \{0,1\}$. The *equality* target function is
$$a(u_1,u_2) \triangleq \begin{cases} 1, & \text{if } u_1 = u_2 \\ 0, & \text{otherwise} \end{cases}$$
for all $u_1, u_2 \in \mathcal{U}$.
- Let $\mathcal{W} = \{0,1\}$. The *greater-than* target function is
$$a(u_1,u_2) \triangleq \begin{cases} 1, & \text{if } u_1 > u_2 \\ 0, & \text{otherwise} \end{cases}$$
for all $u_1, u_2 \in \mathcal{U}$.

We now introduce two channels $g(\cdot,\cdot)$.

- Let $\mathcal{X} = \{0,1\}$ and $\mathcal{Y} = \{0,1,2\}$. The *binary adder* MAC is given by
$$g(x_1,x_2) \triangleq x_1 + x_2$$
for all $x_1, x_2 \in \mathcal{X}$, and where $+$ denotes ordinary addition.
- Let $\mathcal{X} = \{0,1\}$ and $\mathcal{Y} = \{0,1\}$. The *Boolean* $\vee$ or *Boolean OR* MAC is
$$g(x_1,x_2) \triangleq \begin{cases} 0, & \text{if } x_1 = x_2 = 0 \\ 1, & \text{otherwise} \end{cases}$$
for all $x_1, x_2 \in \mathcal{X}$.

$\lozenge$

The emphasis in this paper is on the asymptotic behavior for large function domains, i.e., $U \to \infty$. We allow the other cardinalities $X(U)$, $Y(U)$ and $W(U)$ to scale as a function of U. We use the notation

$$X(U) \dot{\leq} U^a$$

for the relation

$$\limsup_{U\to\infty} \frac{\log(X(U))}{\log(U)} \leq a$$

and analogously for $\dot{<}$. Similarly, we use

$$X(U) \dot{\geq} U^a$$

for the relation

$$\liminf_{U\to\infty} \frac{\log(X(U))}{\log(U)} \geq a$$

and analogously for $\dot{>}$. Finally,

$$X(U) \doteq U^a$$

is short hand for

$$X(U) \dot{\leq} U^a \quad \text{and} \quad X(U) \dot{\geq} U^a.$$

For example, $X(U) \doteq U^a$ is equivalent to $X(U) = U^{a \pm o(1)}$ as $U \to \infty$. With slight abuse of notation, we will write $X(U) \dot{<} U^\infty$ to mean that $X(U) \dot{\leq} U^\eta$ for *some* finite η.

Throughout this paper, we are interested in efficient computation of the target function $a(\cdot,\cdot)$ over the channel $g(\cdot,\cdot)$. In Theorems 1 and 2 only a single use of the channel is permitted, and efficiency is expressed in terms of the required cardinalities $X(U)$ and $Y(U)$ of the channel alphabets as functions of U. In Theorems 3 and 4, multiple uses of the channel are allowed, and efficiency is then naturally expressed in terms of the number of required channel uses $n(U)$ as a function of U.

Finally, all results are stated in terms of the fraction of channels (in Theorems 1 and 2) or target functions (in Theorem 4) for which successful computation is possible.

III. Main Results

Let $\boldsymbol{A}_I \in \mathcal{A}(U, U^2)$ be the identity target function introduced in Example 1, and let $\boldsymbol{G}$ be an arbitrary channel matrix. Consider any other target function $\boldsymbol{A} \in \mathcal{A}(U, W)$ over the same domain $\mathcal{U} \times \mathcal{U}$, but with possibly different range $\mathcal{W}$. Assume $(\boldsymbol{A}_I, \boldsymbol{G})$ is δ-feasible. Then $(\boldsymbol{A}, \boldsymbol{G})$ is also δ-feasible, since we can first compute $\boldsymbol{A}_I$ (and hence $\widehat{\mathsf{u}}_1$ and $\widehat{\mathsf{u}}_2$) over the channel $\boldsymbol{G}$ and then simply apply the function $\boldsymbol{A}$ to the recovered messages $\widehat{\mathsf{u}}_1$ and $\widehat{\mathsf{u}}_2$. This architecture, separating the computation task from the communication task, is illustrated in Fig. 2.

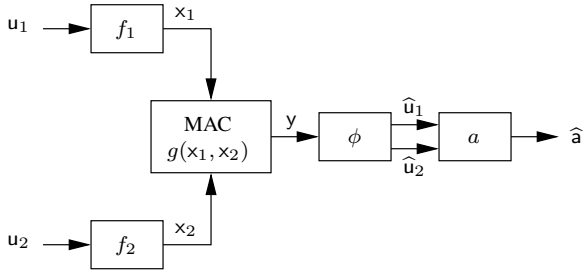

Fig. 2. Separation-based scheme computing the function $a(\cdot,\cdot)$ over the MAC $g(\cdot,\cdot)$. The receiver first decodes the original messages $(\widehat{\mathsf{u}}_1, \widehat{\mathsf{u}}_2)$ and then evaluates the desired target function $a(\widehat{\mathsf{u}}_1, \widehat{\mathsf{u}}_2)$.

As a concrete example, let $\boldsymbol{A}$ be the greater-than target function introduced in Example 1. The range $\mathcal{W} = \{0, 1\}$ of $\boldsymbol{A}$ has cardinality two. On the other hand, the identity function $\boldsymbol{A}_I$ has range $\mathcal{U} \times \mathcal{U}$ of cardinality U^2. In other words, for large U, the identity target function is considerably more complicated than the greater-than target function. As a result, one might expect that the separation-based architecture in Fig. 2 is highly suboptimal in terms of the computation efficiency as described in Section II. As the main result of this paper, we prove that this intuition is wrong in most cases. Instead, we show that for most pairs $(\boldsymbol{A}, \boldsymbol{G})$ of target function and MAC, separation of computation and communication is close to optimal.

We discuss the single channel-use case in Section III-A, and the n channel-uses case in Section III-B.

A. Single Channel Use ($n = 1$)

In this section, we will focus on the case where the target function needs to be computed using just one use of the channel. In this case, the natural value of the upper bound on the probability of error is $\delta = 0$. In other words, we will be interested in 0-feasibility.

We start by deriving conditions under which computation of the identity target function (also introduced in Example 1) over a MAC is feasible. Equivalently, these conditions guarantee that *any* target function with same domain cardinality U can be computed over a MAC by separating computation and communication as discussed above.

Theorem 1. *Let $\boldsymbol{A}_I \in \mathcal{A}(U, U^2)$ be the identity target function, and assume*

$$X(U) \dot{>} U. \tag{1a}$$

If the MAC $\boldsymbol{G} \in \mathcal{G}(X(U), Y(U))$ contains at least $X^2(U) - X(U) + 1$ distinct entries, then $(\boldsymbol{A}_I, \boldsymbol{G})$ is 0-feasible.

In particular, if in addition

$$Y(U) \dot{>} U^3, \tag{1b}$$

then

$$\lim_{U \to \infty} \frac{\left|\left\{\boldsymbol{G} \in \mathcal{G}(X(U), Y(U)) : (\boldsymbol{A}_I, \boldsymbol{G}) \text{ is 0-feasible}\right\}\right|}{|\mathcal{G}(X(U), Y(U))|} = 1.$$

Recall that $\mathcal{G}(X, Y)$ is the collection of all channels $\boldsymbol{G}$ of dimension $X \times X$ and range of cardinality Y. Theorem 1 (together with the separation approach discussed earlier) thus imply roughly that any target function of domain of cardinality U can be computed over most MACs of input cardinality $X(U)$ of order at least U and output cardinality $Y(U)$ of order at least U^3. The precise meaning of "most" is that the fraction of channels $\boldsymbol{G}$ for which the statement holds goes to one as $U \to \infty$. In fact, the fraction of channels for which the theorem fails to hold is exponentially small.

It follows trivially that the above result also holds for any arbitrary target function $\boldsymbol{A} \in \mathcal{A}(U, W(U))$, regardless of the range cardinality $W(U)$. Similarly, since it is clear that the channel input alphabet cardinality $X(U)$ has to be at least of order U for successful computation, we see that the condition on $X(U)$ in Theorem 1 is not a significant restriction. What is significant, however, is the restriction that $Y(U)$ is at least of order U^3. The next result shows that this restriction on $Y(U)$ is essentially also necessary.

Before we state the theorem, we need to introduce one more concept.

Definition. Consider a target function $a \colon \mathcal{U} \times \mathcal{U} \to \mathcal{W}$. For a set $\mathcal{W}_i \subset \mathcal{W}$, consider

$$a^{-1}(\mathcal{W}_i) \triangleq \{(u_1, u_2) \in \mathcal{U} \times \mathcal{U} : a(u_1, u_2) \in \mathcal{W}_i\}.$$

For $c \in (0, 1/2]$, the target function $a(\cdot,\cdot)$ is said to be c-*balanced* if there exist a partition $\mathcal{W}_0, \mathcal{W}_1$ of $\mathcal{W}$ such that

$$|a^{-1}(\mathcal{W}_i)| \geq c \cdot U^2$$

for all $i \in \{0, 1\}$.

Most functions are c-balanced for any $c < 1/3$ and $W(U)$ as long as U is large enough. Indeed, choosing $\mathcal{W}_1 = \{0, \ldots, \lfloor W(U)/2 \rfloor - 1\}$ and $\mathcal{W}_2 = \{\lfloor W(U)/2 \rfloor, \ldots, W(U) - 1\}$ shows that

$$\lim_{U\to\infty} \frac{\left|\left\{\boldsymbol{A} \in \mathcal{A}(U, W(U)) : \boldsymbol{A} \text{ is } 1/3\text{-balanced}\right\}\right|}{|\mathcal{A}(U, W(U))|} = 1, \quad (2)$$

where we recall that $\mathcal{A}(U, W)$ denotes the collection of all target functions $\boldsymbol{A}$ of dimension $U \times U$ and range of cardinality W. In fact, the convergence in (2) is again exponentially fast[2] in U. Moreover, many functions of specific interest are balanced.

Example 2. Consider the target functions introduced in Example 1.

- The identity and the greater-than target functions are c-balanced for any constant $c < 1/2$ and U large enough.
- The equality target function is *not* c-balanced for any constant $c > 0$ as $U \to \infty$. Indeed, since $W(U) = 2$ in this case, the only choice (up to labeling) is to set $\mathcal{W}_0 = \{0\}$ and $\mathcal{W}_1 = \{1\}$. Then $|a^{-1}(\mathcal{W}_0)| = U^2 - U$ and $|a^{-1}(\mathcal{W}_1)| = U$, which is not c-balanced for any constant $c > 0$ as $U \to \infty$.

$\lozenge$

We have the following converse result to Theorem 1 for balanced target functions.

Theorem 2. *Fix a constant $c \in (0, 1/2]$ independent of U. Assume $W(U) \geq 2$ and*

$$X(U) \dot{<} U^{\infty}, \quad (3a)$$

$$Y(U) \dot{<} U^3. \quad (3b)$$

Let $\boldsymbol{A} \in \mathcal{A}(U, W(U))$ be any c-balanced target function. Then

$$\lim_{U\to\infty} \frac{\left|\left\{\boldsymbol{G} \in \mathcal{G}(X(U), Y(U)) : (\boldsymbol{A}, \boldsymbol{G}) \text{ is } 0\text{-feasible}\right\}\right|}{|\mathcal{G}(X(U), Y(U))|} = 0.$$

Recall that the notation $X(U) \dot{<} U^{\infty}$ is used to indicate that $X(U)$ grows at most polynomially in U—an assumption that is quite mild. Thus, Theorem 2 states roughly that in order to compute any balanced target function over most MACs, the cardinality $Y(U)$ of the channel output has to be at least of order U^3 regardless of the value of $W(U)$. Here the precise meaning of "most" is again that the fraction of channel matrices for which computation with fewer channel outputs $Y(U)$ is possible converges to zero, and again this convergence is, in fact, exponentially fast in U. Moreover, combined with (2), this shows that for most pairs $(\boldsymbol{A}, \boldsymbol{G})$ of target and channel functions, the cardinality of the channel output has to be at least of order U^3 for successful computation.

Comparing this to Theorem 1, we see that the same scaling of $Y(U)$ allows computation of a target function using a separation based scheme (i.e., by first recovering the two messages $(\widehat{\mathsf{u}}_1, \widehat{\mathsf{u}}_2)$ and then applying the target function to compute the estimate $\widehat{\mathsf{a}} = a(\widehat{\mathsf{u}}_1, \widehat{\mathsf{u}}_2)$). Thus, for the computation of a given balanced function over most MACs, separation of computation and communication is essentially optimal. Moreover, since most functions are balanced by (2), the same also holds for most pairs $(\boldsymbol{A}, \boldsymbol{G})$ of target and channel functions.

Example 3. Let $\boldsymbol{A}$ be the greater-than target function of domain $U \times U$ introduced in Example 1. Note that this target function has range of cardinality $W(U) = 2$, i.e., $\boldsymbol{A}$ is binary. From Example 2, we know that $\boldsymbol{A}$ is balanced for any constant $c < 1/2$ and U large enough. Thus Theorem 2 applies, showing that, for large U and most MACs $\boldsymbol{G}$, separation of computation and communication is essentially optimal.

Observe that the receiver is interested in only a *single* bit of information about $(\mathsf{u}_1, \mathsf{u}_2)$. Nevertheless, the structure of the greater-than target function is complicated enough that, in order to recover this single bit, the decoder is essentially forced to learn $(\mathsf{u}_1, \mathsf{u}_2)$ itself. In other words, in order to compute the single desired bit, communication of $2\log(U)$ message bits are essentially necessary. $\lozenge$

Theorem 2 is restricted to balanced functions. Even though only a vanishingly small fraction of target functions is not balanced, it is important to understand this restriction. We illustrate this through the following example.

Example 4. Assume $W(U) = 2$ and

$$X(U) \dot{>} U, \quad (4a)$$

$$Y(U) \dot{>} U. \quad (4b)$$

Let $\boldsymbol{A}_{=} \in \mathcal{A}(U, 2)$ be the equality target function introduced in Example 1. Then[3]

$$\lim_{U\to\infty} \frac{\left|\left\{\boldsymbol{G} \in \mathcal{G}(X(U), Y(U)) : (\boldsymbol{A}_{=}, \boldsymbol{G}) \text{ is } 0\text{-feasible}\right\}\right|}{|\mathcal{G}(X(U), Y(U))|} = 1. \quad (5)$$

This result shows that the equality function can be computed over a large fraction[4] of MACs with output cardinality $Y(U)$ of order at least U. This contrasts with output cardinality $Y(U)$ of order U^3 that is required for successful computation of balanced functions in Theorem 2. Recall from Example 2 that the equality target function is *not* c-balanced for any $c > 0$ and U large enough. Thus, (5) does not contradict Theorem 2. It does, however, show

[2]This follows directly from results on the convergence of empirical distributions.

[3]Similar to Theorem 1, one can state precise conditions for an arbitrary $\boldsymbol{G}$ to enable computation of the equality target function. We again show that most $\boldsymbol{G}$ satisfy these conditions. Since these conditions are a bit more complicated, we do not state them here, but refer the reader to [15] instead.

[4]As an aside, the proof only shows that this fraction converges to 1 inversely in U and therefore the fraction of "bad" channels for computing the equality function diminishes much slower than those in Theorems 1 and 2.

that for unbalanced functions separation of computation and communication can be suboptimal. $\lozenge$

B. Multiple Channel Uses ($n \geq 1$)

In this section, we allow multiple uses of the MAC. Our emphasis will again be on the asymptotic behavior for large function domains $U \to \infty$. However, in this section we keep the MAC $g(\cdot,\cdot)$, and hence also the cardinalities of the channel domain $\mathcal{X}$ and channel range $\mathcal{Y}$, fixed. Instead, we characterize the minimum number $n = n(U)$ of channel uses required to compute the target function.

As we shall see, the behavior of $n(U)$ is governed by the maximum entropy that can be induced at the channel output. Formally, for a MAC $g(\cdot,\cdot)$, define

$$H^\star(g) \triangleq \max_{\mathsf{x}_1,\mathsf{x}_2} H\big(g(\mathsf{x}_1,\mathsf{x}_2)\big), \tag{6}$$

where the maximization is over all independent random variables $\mathsf{x}_1, \mathsf{x}_2$ taking values in the channel input alphabet $\mathcal{X}$, and where $H(\cdot)$ denotes entropy. When convenient, we use the notation $H^\star(\boldsymbol{G})$ instead of $H^\star(g)$, where $\boldsymbol{G}$ is the channel matrix corresponding to the MAC $g(\cdot,\cdot)$. For simplicity, we will restrict attention to the following class of MACs.

Definition. A MAC $\boldsymbol{G}$ is said to be *equal-rate optimal* if the point $(H^\star(\boldsymbol{G})/2, H^\star(\boldsymbol{G})/2)$ lies in the capacity region of $\boldsymbol{G}$, where the capacity region of a MAC is as defined in [16, Theorem 14.3.1].

Example 5. Both the binary adder MAC and the Boolean OR MAC introduced in Example 1 are equal-rate optimal. In general, any symmetric MAC is equal-rate optimal. $\lozenge$

We begin by stating a result for the identity target function introduced in Example 1. Equivalently, this result applies to *any* target function (with the same domain cardinality U) by using a communication scheme separating computation and communication.

Theorem 3. *Fix a constant $\delta > 0$ independent of U, and assume that X and Y are constant. Let $\boldsymbol{A}_I \in \mathcal{A}(U, U^2)$ be the identity target function, and let $\boldsymbol{G} \in \mathcal{G}(X,Y)$ be any equal-rate optimal MAC. Then, for any $n(U)$ such that*

$$2^{n(U)H^\star(\boldsymbol{G})} > U^2,$$

$(\boldsymbol{A}_I, \boldsymbol{G}^{\otimes n(U)})$ is δ-feasible for U large enough.

The result follows directly from the maximum achievable sum-rate for ordinary communication over the MAC, see for example [16, Theorem 14.3.1], and the assumption that the MAC $\boldsymbol{G}$ is equal-rate optimal.

Using separation, Theorem 3 implies that for large enough U, any target function of cardinality U can be reliably computed over $n(U)$ uses of an equal-rate optimal MAC $\boldsymbol{G}$ as long as $2^{n(U)H^\star(\boldsymbol{G})}$ is bigger than U^2. Or, equivalently, the number of channel uses $n(U)$ needs to be larger than $2\log(U)/H^\star(\boldsymbol{G})$. The next result states that for most functions this restriction on $n(U)$ is essentially also necessary.

Theorem 4. *Assume that*[5]

$$W(U) \geq \omega(1)$$

as $U \to \infty$, that $0 \leq \delta \leq 1/(2\ln(W(U)))$, and that X and Y are constant. Let $\boldsymbol{G} \in \mathcal{G}(X,Y)$ be any MAC. Then, for any $n(U)$ such that

$$2^{n(U)H^\star(\boldsymbol{G})} \doteq U^2,$$

we have

$$\lim_{U\to\infty} \frac{\big|\big\{\boldsymbol{A} \in \mathcal{A}(U,W(U)) : (\boldsymbol{A}, \boldsymbol{G}^{\otimes n(U)}) \text{ is } \delta\text{-feasible}\big\}\big|}{|\mathcal{A}(U,W(U))|} = 0.$$

Recall that $\mathcal{A}(U,W)$ denotes the collection of all target functions $\boldsymbol{A}$ of dimension $U \times U$ and range of cardinality W. Together, Theorems 3 and 4 thus show that, for any equal-rate optimal MAC and most target functions, the smallest number of channel uses $n^\star(U)$ that enables reliable computation is of order $2\log(U)/H^\star(\boldsymbol{G})$. Moreover, they show that for most such pairs, separation of computation and communication is essentially necessary and sufficient if we allow multiple uses of the channel and nonzero error probability. Here the precise meaning of "most" is that the statement holds for all but a vanishing fraction of functions. Moreover, the proof of the theorem shows again that this fraction is, in fact, exponentially small in U.

Example 6. Consider a target function of range with cardinality $W(U) = \log(U)$. Let $\boldsymbol{G}$ be the binary adder MAC introduced in Example 1. Then

$$H^\star(\boldsymbol{G}) = 3/2.$$

Theorems 3 and 4 show that, for most functions $\boldsymbol{A}$ of domain $U \times U$ and range $\log(U)$, the smallest number of channel uses $n^\star(U)$ required for reliable computation is of order $4\log(U)/3$. Moreover, this performance can be achieved by separating computation and communication. In other words, even though the receiver is only interested in $\log\log(U)$ function bits, it is essentially forced to learn the $2\log(U)$ message bits as well.

This example also illustrates that the usual way of proving converse results based on the cut-set bound is not tight for most $(\boldsymbol{A}, \boldsymbol{G})$ pairs. For example, [11, Lemma 13] shows using the cut-set bound that for reliable computation we need to have

$$n(U)H^\star(\boldsymbol{G}) \geq H(a(\mathsf{u}_1,\mathsf{u}_2))$$

where $H(\cdot)$ denotes entropy. Since $\boldsymbol{A}$ has range of cardinality $W(U)$, we have

$$H(a(\mathsf{u}_1,\mathsf{u}_2)) \leq \log(W(U)).$$

For $W(U) = \log(U)$ and $H^\star(\boldsymbol{G}) = 3/2$ as considered here, the tightest bound that can in the *best case* be derived via the cut-set approach is thus

$$n^\star(U) \geq \log(W(U))/H^\star(\boldsymbol{G}) = 2\log\log(U)/3.$$

[5]Recall that the notation $W(U) \geq \omega(1)$ as $U \to \infty$ stands for $\lim_{U\to\infty} W(U) = \infty$.

However, we know that the correct scaling for $n^\star(U)$ is $4\log(U)/3$. Hence, the cut-set bound is loose by an unbounded factor as $U \to \infty$. $\Diamond$

REFERENCES

[1] E. Kushilevitz and N. Nisan, *Communication Complexity*. Cambridge University Press, 2006.

[2] A. C.-C. Yao, "Some complexity questions related to distributive computing," in *Proc. ACM STOC*, 1979, pp. 209–213.

[3] J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 219–221, Mar. 1979.

[4] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 903–917, Mar. 2001.

[5] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 755–764, Apr. 2005.

[6] V. Doshi, D. Shah, M. Médard, and M. Effros, "Functional compression through graph coloring," *IEEE Trans. Inf. Theory*, vol. 56, no. 8, pp. 3901–3917, Aug. 2010.

[7] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, "Network coding for computing: Cut-set bounds," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1015–1030, Feb. 2011.

[8] T. M. Cover, A. El Gamal, and M. Salehi, "Multiple access channels with arbitrarily correlated sources," *IEEE Trans. Inf. Theory*, vol. 26, no. 6, pp. 648– 657, Nov. 1980.

[9] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proc. ACM MobiCom*, Sep. 2006, pp. 358–365.

[10] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," in *Proc. ACM SIGCOMM*, Oct. 2007, pp. 397–408.

[11] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, Oct. 2007.

[12] M. P. Wilson, K. Narayanan, H. D. Pfister, and A. Sprintson, "Joint physical layer coding and network coding for bidirectional relaying," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5641–5654, Nov. 2010.

[13] U. Niesen and P. Whiting, "The degrees of freedom of compute-and-forward," *arXiv:1101.2182 [cs.IT]*, Jan. 2011, to appear in IEEE Trans. Inf. Theory.

[14] L. Keller, N. Karamchandani, and C. Fragouli, "Function computation over linear channels," in *Proc. IEEE NetCod*, Jun. 2010, pp. 1–6.

[15] N. Karamchandani, U. Niesen, and S. Diggavi, "Computation over mismatched channels," *pre-print*, 2012. [Online]. Available: http://arxiv.org/abs/1204.5059

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 1991.